



(19)

(11) Publication number:

05334055 A

Generated Document

PATENT ABSTRACTS OF JAPAN

(21) Application number: 04276859

(51) Intl. Cl.: G06F 9/06 G06F 9/45

(22) Application date: 15.10.92

(30) Priority: 25.10.91 JP 03279549

(43) Date of application publication: 17.12.93

(84) Designated contracting states:

(71) Applicant: NEC IC MICROCOMPUT SYST LTD

(72) Inventor: TAKAGI HIDEHIKO

(74) Representative:

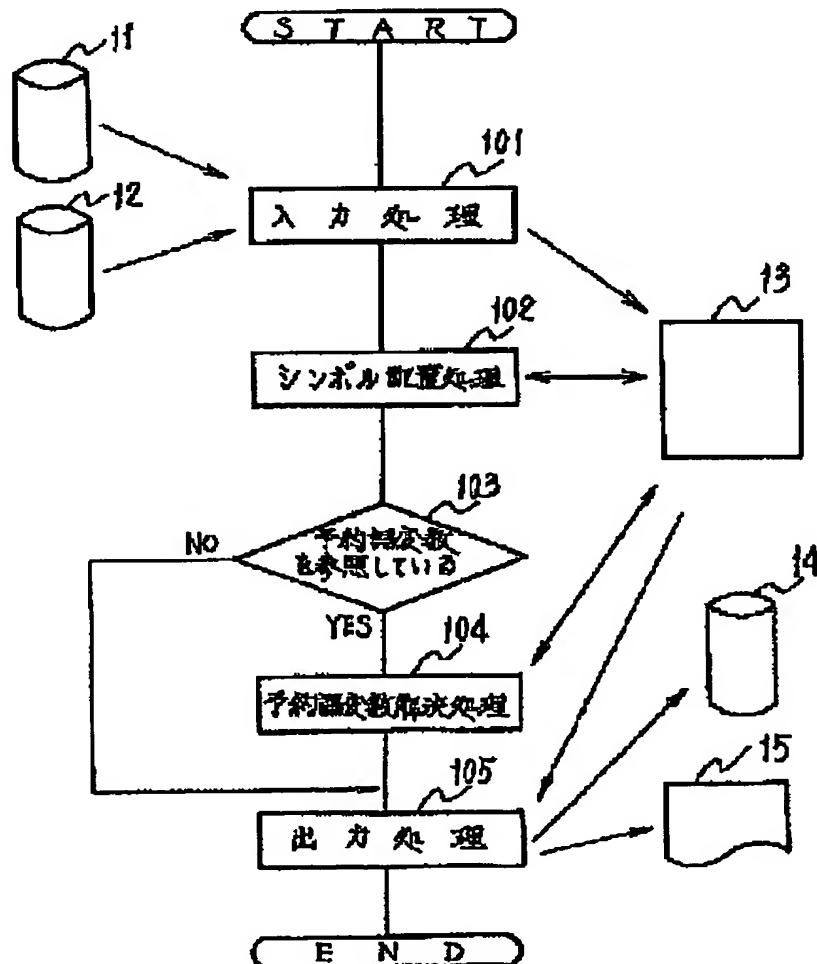
(54) LINKER

(57) Abstract:

PURPOSE: To easily reserve a maximum data area by inserting the start address and the size of a remaining area to the positions of two reserved word variables.

CONSTITUTION: In the discrimination processing, it is checked whether a reserved word variable 1 indicating the start address of a maximum continuous unused area in a data area and a reserved word variable 2 indicating the size of this maximum area, are referred to or not. When they are referred to, the reserved word variable resolution processing is performed, and a symbol table 13 is referred to perform the resolution processing of addresses to reserved word variables 1 and 2. If reserved word variables 1 and 2 are not referred to in the discrimination processing, a completed machine word is outputted as an executable program 14, and the symbol table 13 is outputted as a map list 15 as information of the arrangement result.

COPYRIGHT: (C)1993,JPO&Japio



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-334055

(43) 公開日 平成5年(1993)12月17日

(51) Int.Cl.⁵

G 0 6 F 9/06
9/45

識別記号

4 1 0 E

庁内整理番号

7232-5B

9292-5B

F 1

技術表示箇所

G 0 6 F 9/44

3 2 2 K

審査請求 未請求 請求項の数3(全11頁)

(21) 出願番号 特願平4-276859

(22) 出願日 平成4年(1992)10月15日

(31) 優先権主張番号 特願平3-279549

(32) 優先日 平3(1991)10月25日

(33) 優先権主張国 日本(J P)

(71) 出願人 000232036

日本電気アイシーマイコンシステム株式会
社

神奈川県川崎市中原区小杉町1丁目403番
53

(72) 発明者 ▲高▼木 秀彦

神奈川県川崎市中原区小杉町一丁目403番
53日本電気アイシーマイコンシステム株式
会社内

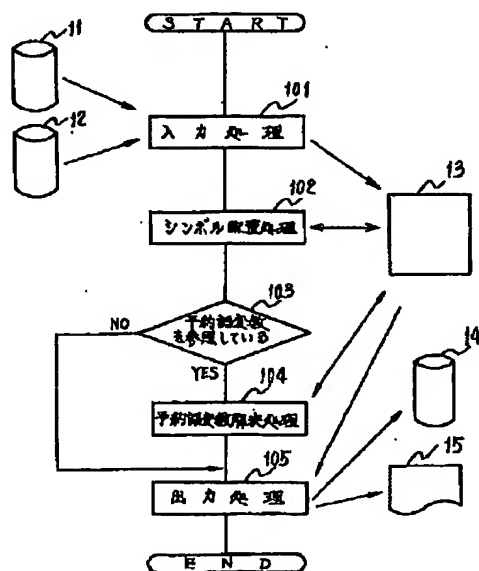
(74) 代理人 弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 リンカ

(57) 【要約】

【目的】 残りデータ領域の大きさを計算することなく、最大限のデータ領域を極めて容易に確保することができるリンカを提供する。

【構成】 本発明のリンカは、入力処理ステップ101と、シンボル配置処理ステップ102と、予約語変数を参照しているか否かの処理ステップ103と、予約語変数解決処理ステップ104と、出力処理ステップ105を有している。



11, 12...オブジェクト 13...シンボルテーブル

14...実行可能プログラム 15...マップ・リスト

1

【特許請求の範囲】

【請求項1】 指定された全てのオブジェクトを入力しながら、所定のシンボル・テーブル内に、原始プログラム内で定義された相互参照用のシンボル名とその実際の位置情報を、個々のオブジェクトの配置情報と、そのオブジェクト内の個々のシンボルの相対位置情報より計算しながら登録を行う第1の処理ステップと、

前記第1の処理ステップにおいて読込まれたオブジェクトの機械語部分の中で、シンボルを参照している機械語部分を、完成されたシンボル・テーブルを参照しながら修正して完全な機械語を生成する第2の処理ステップと、

データ領域中の連続した最大未使用領域の先頭番地を示す予約語変数1と、当該データ領域の大きさを示す予約語変数2が参照されているか否かを判定する第3の処理ステップと、

前記予約語変数1および2が参照されている場合に、当該予約語変数1および2に対応する番地を、前記シンボル・テーブルを参照して解決する第4の処理ステップと、

前記第4の処理ステップによる番地の解決後、または前記第3の処理ステップにおいて、前記予約語変数1および2が参照されていない場合に、完成された機械語を実行可能プログラムとして出力するとともに、配置結果の情報として、前記シンボル・テーブルをマップ・リストとして出力する第5の処理ステップと、

を有することを特徴とするリンク。

【請求項2】 前記第4の処理ステップにおける処理手順として、

未使用データ領域における連続した最大領域の「大きさ」および「先頭番地」をそれぞれX₁およびP₁に設定し、未使用データ領域における連続した2番目に大きい領域の「大きさ」および「先頭番地」をそれぞれX₂およびP₂に設定する第6の処理ステップと、

予約語変数2の「大きさ」と前記X₂の「大きさ」とを比較判定する第7の処理ステップと、

前記第7の処理ステップにおいて、X₂の「大きさ」が予約語変数2の「大きさ」以上である場合に、予約語変数2の番地として前記P₂を設定する第8の処理ステップと、

前記第7の処理ステップにおいて、X₂の「大きさ」が予約語変数2の「大きさ」より小さい場合に、予約語変数2の「大きさ」と前記X₁の「大きさ」とを比較判定する第9の処理ステップと、

前記第9の処理ステップにおいて、X₁の「大きさ」が予約語変数2の「大きさ」以上である場合に、予約語変数2の番地として前記P₁を設定するとともに、前記P₁には予約語変数2の「大きさ」を加算して、X₁からは予約語変数2の「大きさ」を減算する第10の処理ステップと、

2

前記第8の処理ステップならびに前記第10の処理ステップにフォローして、X₁の「大きさ」と予約語変数1の1配列当りの「大きさ」とを比較判定する第11の処理ステップと、

前記第11の処理ステップにおいて、X₁の「大きさ」が予約語変数1の1配列当りの「大きさ」以上である場合に、予約語変数2の番地としてP₁を設定する第12の処理ステップと、

10 前記第11の処理ステップにおいて、X₁の「大きさ」が予約語変数1の1配列当りの「大きさ」より小さい場合、ならびに前記第9の処理ステップにおいて、X₁の「大きさ」が予約語変数2の「大きさ」より小さい場合に、空きデータ領域不足のエラー・メッセージを出力して処理を終了する第13の処理ステップと、

前記第12の処理ステップにフォローして、予約語変数2の初期値としてX₁の予約語変数1の1配列当りの「大きさ」で除した値を設定して処理を終了する第14の処理ステップと、

を有することを特徴とする請求項1記載のリンク。

20 【請求項3】 前記第4の処理ステップにおける処理手順として、

未使用データ領域における連続した最大領域の「大きさ」および「先頭番地」をそれぞれX₁およびP₁に設定し、未使用データ領域に空き領域がない場合には「0」を設定する第15の処理ステップと、

予約語変数2の初期値の「大きさ」と前記X₁の「大きさ」とを比較判定する第16の処理ステップと、

前記第16の処理ステップにおいて、X₁の「大きさ」が予約語変数2の初期値の「大きさ」以上である場合に、予約語変数1の番地として前記P₁を設定する第17の処理ステップと、

予約語変数2に格納されている値をX₁に書換えて処理を終了する第18の処理ステップと、

前記第16の処理ステップにおいて、X₁の「大きさ」が予約語変数2の「大きさ」より小さい場合に、空きデータ不足のエラー・メッセージを出力して処理を終了する第19の処理ステップと、

を有することを特徴とする請求項1記載のリンク。

【発明の詳細な説明】

40 【0001】

【産業上の利用分野】 本発明はリンクに関する。

【0002】

【従来の技術】 従来のリンクの動作例について、図5に示されるフローチャートを参照して説明する。

40 【0003】 図5において、まず、ステップ501において、指定された全てのオブジェクト51および52を入力しながら、シンボル・テーブル53内に原始プログラム内で定義されている相互参照用のシンボル名とその実際の位置情報を、個々のオブジェクトの配置情報と、そのオブジェクト内の個々のシンボルの相対位置情報よ

り計算しながら登録を行う。ここで、シンボル・テーブル53とは、シンボル名と、そのシンボル名の実際の位置情報を対にして保有する表のことであり、以後の再配置を制御するために必要なものである。

【0004】次に、ステップ502において、読込まれたオブジェクトの機械語部分の中において、シンボルを参照している機械語部分を、完成されたシンボル・テーブル53を参照しながら修正して完全な機械語を生成する。そして、最後に、ステップ503において、完成された機械語を実行可能プログラム54として出力するとともに、最終配置結果を設計者に伝達するために、当該配置結果の情報としてシンボル・テーブルをマップ・リスト55として出力する。

【0005】しかしながら、本発明の対象であるリンカおよび一般の言語処理プログラムの場合には、前述したシンボル・テーブルのように、出来得る限り大量に確保した方が有利な1変数域を有するという特徴がある。一般に、データを大量に入力して、その相互関係を処理するプログラムの場合には、このような傾向がある。

【0006】上述のようなプログラムを開発する場合には、従来のリンカを使用するプログラム開発環境においては、最初に前記の1変数域を仮の量で確保して1度プログラムを作成し、前記マップ・リスト55より確保できるメモリ容量を逆算して、その値で最初の原始プログラムを修正し、最終の機械語を生成する必要がある。

【0007】図6に示されるのは、このプログラム開発手順の流れを示す図であり、残りデータ領域を割当てて配列以外の変数の定義を行った原始プログラム61および62を、機械語翻訳プログラム63を介して翻訳してオブジェクト64を得る。次いで、リンカ65にオブジェクト64を入力して、実行可能プログラム66とマップ・リスト67を作成する。このマップ・リスト67からプログラムが参照を許される残りの領域を調べて、このうち連続している最大領域の大きさを手計算により求める。次に、処理ステップ601において、原始プログラム61および62において求めた最大領域の大きさの配列を定義して、機械語翻訳プログラム63に再入力することにより、データ領域を最大限に利用するのが一般である。

【0008】

【発明が解決しようとする課題】上述した従来のリンカにおいては、データ領域を最大限に使用するために、必要最小限のデータの定義後において、残りデータ領域を1配列として使用しようとする場合に、機械語翻訳プログラム自身が使用するデータ領域と自分自身で定義したデータ領域から、残りデータ領域を手計算により求めなければならない。

【0009】特に、機械語翻訳プログラム自身が使用するデータ領域は、1度リンカを起動してリンカの出力するマップ・リストにより算出しなければならず、従っ

て、残りデータ領域を算出するためには、多大な手間を要するという欠点がある。しかも算出した数字に誤謬があった場合には、リンカでのエラーを確認後に、再度計算をやり直さなければならないという欠点がある。

【0010】

【課題を解決するための手段】本発明のリンカは、指定された全てのオブジェクトを入力しながら、所定のシンボル・テーブル内に、原始プログラム内で定義された相互参照用のシンボル名とその実際の位置情報を、個々のオブジェクトの配置情報と、そのオブジェクト内の個々のシンボルの相対位置情報より計算しながら登録を行う第1の処理ステップと、前記第1の処理ステップにおいて読込まれたオブジェクトの機械語部分の中で、シンボルを参照している機械語部分を、完成されたシンボル・テーブルを参照しながら修正して完全な機械語を生成する第2の処理ステップと、データ領域中の連続した最大未使用領域の先頭番地を示す予約語変数1と、当該データ領域の大きさを示す予約語変数2が参照されているかを判定する第3の処理ステップと、前記予約語変数1および2が参照されている場合に、当該予約語変数1および2に対応する番地を、前記シンボル・テーブルを参照して解決する第4の処理ステップと、前記第4の処理ステップによる番地の解決後、または前記第3の処理ステップにおいて、前記予約語変数1および2が参照されていない場合に、完成された機械語を実行可能プログラムとして出力するとともに、配置結果の情報として、前記シンボル・テーブルをマップ・リストとして出力する第5の処理ステップと、を有することを特徴としている。

【0011】なお、前記第4の処理ステップにおける処理手順としては、未使用データ領域における連続した最大領域の「大きさ」および「先頭番地」をそれぞれX₁およびP₁に設定し、未使用データ領域における連続した2番目に大きい領域の「大きさ」および「先頭番地」をそれぞれX₂およびP₂に設定する第6の処理ステップと、予約語変数2の「大きさ」と前記X₁の「大きさ」とを比較判定する第7の処理ステップと、前記第7の処理ステップにおいて、X₂の「大きさ」が予約語変数2の「大きさ」以上である場合に、予約語変数2の番地として前記P₂を設定する第8の処理ステップと、前記第7の処理ステップにおいて、X₂の「大きさ」が予約語変数2の「大きさ」より小さい場合に、予約語変数2の「大きさ」と前記X₁の「大きさ」とを比較判定する第9の処理ステップと、前記第9の処理ステップにおいて、X₁の「大きさ」が予約語変数2の「大きさ」以上である場合に、予約語変数2の番地として前記P₁を設定するとともに、前記P₁には予約語変数2の「大きさ」を加算して、X₁からは予約語変数2の「大きさ」を減算する第10の処理ステップと、前記第8の処理ステップならびに前記第10の処理ステップにフォローし

て、 X_1 の「大きさ」と予約語変数 1 の 1 配列当りの「大きさ」とを比較判定する第 11 の処理ステップと、前記第 11 の処理ステップにおいて、 X_1 の「大きさ」が予約語変数 1 の 1 配列当りの「大きさ」以上である場合に、予約語変数 2 の番地として P_1 を設定する第 12 の処理ステップと、前記第 11 の処理ステップにおいて、 X_1 の「大きさ」が予約語変数 1 の 1 配列当りの「大きさ」より小さい場合、ならびに前記第 9 の処理ステップにおいて、 X_1 の「大きさ」が予約語変数 2 の「大きさ」より小さい場合に、空きデータ領域不足のエラー・メッセージを出力して処理を終了する第 13 の処理ステップと、前記第 12 の処理ステップにフォローして、予約語変数 2 の初期値として X_1 の予約語変数 1 の 1 配列当りの「大きさ」で除した値を設定して処理を終了する第 14 の処理ステップとを有してもよい。

【0012】更にまた、前記第 4 の処理ステップにおける処理手順としては、未使用データ領域における連続した最大領域の「大きさ」および「先頭番地」をそれぞれ X_1 および P_1 に設定し、未使用データ領域に空き領域がない場合には「0」を設定する第 15 の処理ステップと、予約語変数 2 の初期値の「大きさ」と前記 X_1 の「大きさ」とを比較判定する第 16 の処理ステップと、前記第 16 の処理ステップにおいて、 X_1 の「大きさ」が予約語変数 2 の初期値の「大きさ」以上である場合に、予約語変数 1 の番地として前記 P_1 を設定する第 17 の処理ステップと、予約語変数 2 に格納されている値を X_1 に書換えて処理を終了する第 18 の処理ステップと、前記第 16 の処理ステップにおいて、 X_1 の「大きさ」が予約語変数 2 の「大きさ」より小さい場合に、空きデータ不足のエラー・メッセージを出力して処理を終了する第 19 の処理ステップとを有するようにしてもよい。

【0013】

【実施例】次に、本発明について図面を参照して説明する。

【0014】図 1 は本発明の基本的な処理手順を示すフローチャートである。図 1 において、まず、ステップ 101 の入力処理において、指定された全てのオブジェクト 11 および 12 を入力しながら、シンボル・テーブル 13 内に原始プログラム内で定義された相互参照用のシンボル名とその実際の位置情報を、個々のオブジェクトの配置情報と、そのオブジェクト内の個々のシンボルの相対位置情報より計算しながら登録を行う。次に、ステップ 102 のシンボル配置処理において、読込まれたオブジェクトの機械語部分の中において、シンボルを参照している機械語部分を、完成されたシンボル・テーブル 13 を参照しながら修正して完全な機械語を生成する。このステップ 101 および 102 に示される処理手順については、前述の従来例の場合と同様である。

【0015】次いで、ステップ 103 の判断処理におい

て、データ領域中の連続した最大未使用領域の先頭番地を示す予約語変数（以下、予約語変数 1 と云う）と、その領域の大きさを示す予約語変数（以下、予約語変数 2 と云う）が参照されているか否かが調べられる。ステップ 103 において、予約語変数 1 および予約語変数 2 が参照されている場合には、ステップ 104 の予約語変数解決処理に進み、ステップ 104 において、当該予約語変数 1 および 2 に対応する番地は、シンボル・テーブル 13 を参照して解決処理される。このステップ 104 における番地の解決後、またはステップ 103 の判断処理において前記予約語変数 1 および予約語変数 2 が参照されていない場合には、ステップ 105 の出力処理において、完成された機械語を実行可能プログラム 14 として出力するとともに、配置結果の情報として、シンボル・テーブル 13 をマップ・リスト 15 として出力する。

【0016】図 2 は、本発明の第 1 の実施例のステップ 104 の予約語変数解決処理における処理手順を示すフローチャートである。まず、ステップ 201 において、未使用データ領域のうち、連続している最大領域の大きさと先頭番地とを、それぞれ X_1 と P_1 に設定し、また 2 番目に大きい領域の大きさと先頭番地とを、それぞれ X_2 と P_2 に設定する。この場合、空き領域がない場合には 0 を設定する。次に、ステップ 202 の判断処理において、予約語変数 2 を格納する領域として、2 番目に大きいデータ領域の大きさが十分であるかどうか、 X_2 と予約語変数 2 の大きさが比較判定される。ステップ 202 において、 X_2 の大きさが予約語変数 2 の大きさ以上である場合には、ステップ 203 に進み、予約語変数 2 を格納する領域として 2 番目に大きい空き領域を使用するために、予約語変数 2 の番地として P_2 を設定する。また、ステップ 202 において、 X_2 の大きさが予約語変数 2 の大きさよりも小さい場合には、最大空き領域に格納することができるかどうかを調べるためにステップ 204 に進み、 X_1 の大きさと予約語変数 2 の大きさが比較判定される。ステップ 204 において、 X_1 の大きさが予約語変数 2 の大きさ以上である場合には、ステップ 205 に進み、予約語変数 2 を格納する領域として最大空き領域を使用するために、予約語変数 2 の番地として P_1 を設定し、更に最大空き領域を使用するための後処理として、 P_1 に予約語変数 2 の大きさを加算し、 X_1 から予約語変数 2 の大きさを減算する。また、ステップ 204 の判断処理において、 X_1 の大きさが予約語変数 2 の大きさよりも小さい場合には、ステップ 209 に進み、空きデータ領域不足のエラー・メッセージを出力して処理を終了する。

【0017】予約語変数 2 の番地設定が完了すると、次に、ステップ 206 において、最大空き領域に少なくとも配列が一つ格納することができるかどうか、即ち X_1 と予約語変数 1 の 1 配列当りの大きさを比較する。このステップ 206 の判断処理において、 X_1 の大きさが予

約語変数1の1配列当りの大きさ以上である場合には、ステップ207に進み、最大空き領域に少なくとも配列が一つ以上は格納することができるため、予約語変数2の番地としてP₁を設定する。続いて、ステップ208において、予約語変数2の初期値として、X₁を予約語変数1の1配列当りの大きさで除した値を設定する。また、ステップ206の判断処理において、X₁の値の方が小さい場合にはステップ209に進み、空きデータ領域不足のエラー・メッセージを出力して処理を終了する。

【0018】次に、本発明の第2の実施例のステップ104の予約語変数解決処理における処理手順について説明する。図3は、当該予約語変数解決処理における処理手順を示すフローチャートである。図3において、先ずステップ301において、未使用データ領域の内、連続している最大領域の大きさをX₁に設定し、連続している最大領域の先頭番地をP₁に設定する。この場合、空き領域がない場合には0を設定する。次に、ステップ302において、ユーザが必要最低限の領域を確保することができるかどうか、即ち、X₁の値と予約語変数2の初期値として与えられた値とを比較する。このステップ302の判断処理において、X₁の値の大きさが予約語変数2の初期値として与えられた値以上である場合には、ステップ303に進み、予約語変数1の番地としてP₁を設定する。続いて、ステップ204において、予約語変数2に格納されている値をX₁に置換える。また、ステップ302の判断処理において、X₁の値の方が小さい場合には、ステップ305に進み、ステップ305において空きデータ領域不足のエラー・メッセージを出力して処理を終了する。

【0019】次に、最大64Kバイトのデータ領域を有するコンピュータに配置されるリンカにおいて、必要最小限のデータの定義後に、残りデータ領域を一つの配列に割り当てるという処理において、一般にC言語と呼ばれる原始プログラムで、予約語変数1と予約語変数2の変数名を仮にkey1とkey2とした場合について、図4に示される原始プログラムを例として翻訳する場合につき、上記の第2の実施例におけるステップ104の予約語変数解決処理の内容を説明する。なお、この時のリンカにおいて使用されるデータ領域を、0H番地から0

【0020】図4における予約語変数2の定義401と、予約語変数1の外部参照宣言402により、key1とkey2とを宣言して、配列初期化処理403のように、key1とkey2の2個の要素からなる配列として参照する。このようにして作成された原始プログラムを、図6に示される機械語翻訳プログラム63により

翻訳し、オブジェクト64を得て、リンカ65に入力する。

【0021】前述のように、図1におけるステップ101の入力処理により指定された全てのオブジェクト11および12を入力しながら、シンボル・テーブル13内に原始プログラム内で定義された相互参照用のシンボル名とその実際の位置情報とを、個々のオブジェクトの配置情報と、そのオブジェクト内の個々のシンボルの相対位置情報より計算しながら登録を行う。次に、ステップ102のシンボル配置処理において、読込まれたオブジェクトの機械語部分の中において、シンボルを参照している機械語部分を、完成されたシンボル・テーブル13を参照しながら修正して完全な機械語を生成する。次に、図4の原始プログラムにおいて、key1が参照され、且つkey2が定義されていないか、図1のステップ103の判断処理において調べる。図4の予約語変数2の定義401と、予約語変数1の外部参照宣言402により、key1が参照され、且つkey2が定義されているので、図1におけるステップ104の予約語変数解決処理によって、その番地を解決する。

【0022】図3は、上述のように、第2の実施例における予約語変数解決処理の処理手順を示すフローチャートである。ステップ301により、未使用データ領域の内、連続した最大領域の大きさと先頭番地をそれぞれX₁およびP₁に設定する。本実施例の場合には、1000H番地からEFFFH番地までのD000Hバイトが連続した空き領域である。従って、X₁にはD000Hが、P₁には1000Hがそれぞれ設定される。ステップ302においては、ユーザが必要とする最低限の領域を確保することができるのか、即ち、X₁と図4の予約語変数2の定義401によって与えられたkey2の初期値「100」を比較する。この場合にはX₁の値の方が大きいので、図3におけるステップ303に進み、予約語変数1の番地としてP₁、即ち1000Hを設定する。続いて、ステップ304においては、予約語変数1に格納されている値をX₁即ちD000Hに置換える。そして最後に、図1のステップ105の出力処理において、完成された機械語を実行可能プログラム14として出力するとともに、配置情報として、シンボル・テーブル13をマップ・リスト15として出力する。

【0023】なお、上記の説明においては、一例として言語処理プログラムの内のC言語を使用するものとして説明を行っているが、本発明は、このC言語による場合に限定されるものではなく、他のアセンブリ言語およびFORTRAN言語等による場合においても同様の効果が得られて、本発明の目的が達成されることは言うまでもない。

【0024】

【発明の効果】以上説明したように、本発明は、機械語翻訳プログラムが入力する原始プログラム内に、予め定

義されている二つの予約語変数を記述する処理手順と、機械語の再配置後に、対象とするコンピュータシステムにおいてプログラムが参照することを容認される残り領域の先頭番地と当該領域の大きさを、前記二つの予約語変数の位置に挿入する処理手順を有することにより、残りデータ領域の大きさを計算することなく、最大限のデータ領域を極めて容易に確保することができるという効果がある。

【0025】また、機械語翻訳プログラム自身において使用するデータ領域に関しては、1度リンカを起動して、リンカの出力するマップ・リストにより算出しなければならないが、この残りデータ領域を算出するために要する多大の手間を排除することができるとともに、誤りがあった場合においても、リンクにおけるエラーを確認後に、再度計算をやり直す手間を省くことができるという効果がある。

【図面の簡単な説明】

【図1】本発明における処理手順を示すフローチャートである。

【図2】本発明の第1の実施例における予約語変数解決処理手順を示すフローチャートである。

【図3】本発明の第2の実施例における予約語変数解決処理手順を示すフローチャートである。

【図4】原始プログラムの一例を示す図である。

【図5】従来例における処理手順を示すフローチャートである。

【図6】プログラム開発手順を示す図である。

【符号の説明】

- 11、12、51、52、64 オブジェクト
- 13、53 シンボル・テーブル
- 14、54、66 実行可能プログラム
- 15、55、67 マップ・リスト
- 61、62 原始プログラム
- 63 機械語翻訳プログラム
- 65 リンカ
- 101、501 入力処理
- 102、502 シンボル配置処理
- 103 予約語変数参照判定処理
- 104 予約語変数解決処理
- 105、503 出力処理
- 201、301 未使用データ領域検出処理
- 202、204 予約語変数2の格納領域判定処理
- 203、205 予約語変数2の格納領域設定処理
- 206、302 予約語変数1の格納領域判定処理
- 207、303 予約語変数2の番地設定処理
- 208、304 予約語変数2の初期値設定処理
- 209、305 エラー・メッセージ出力処理
- 401 予約語変数2の定義
- 402 予約語変数1の外部参照宣言
- 403 配列初期化処理
- 601 変数確保量の修正処理

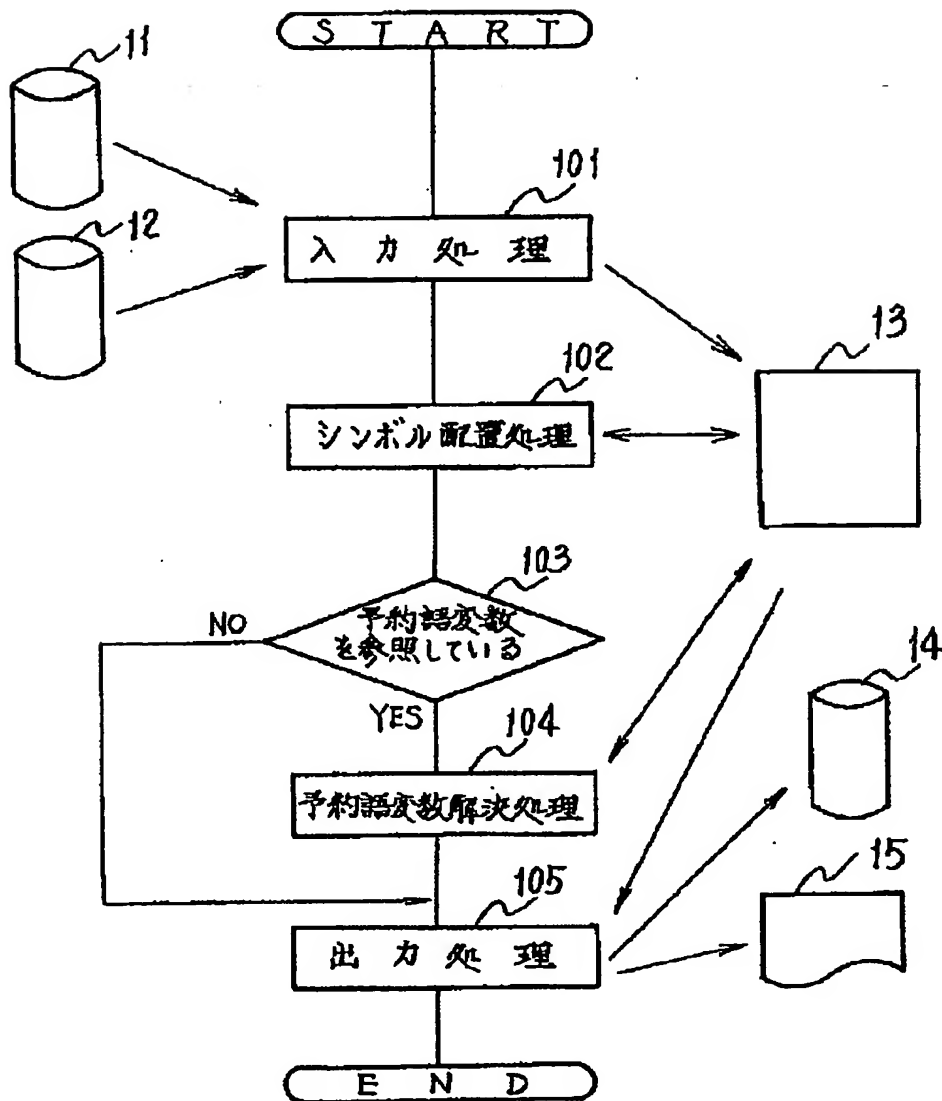
【図4】

```

                                401
                                ~~~~~
int key2=100;
extern char key1[];           402
.
.
for (i=0; i<key2; i++)       403
    key1[i]=0;
.
.

```

【図1】



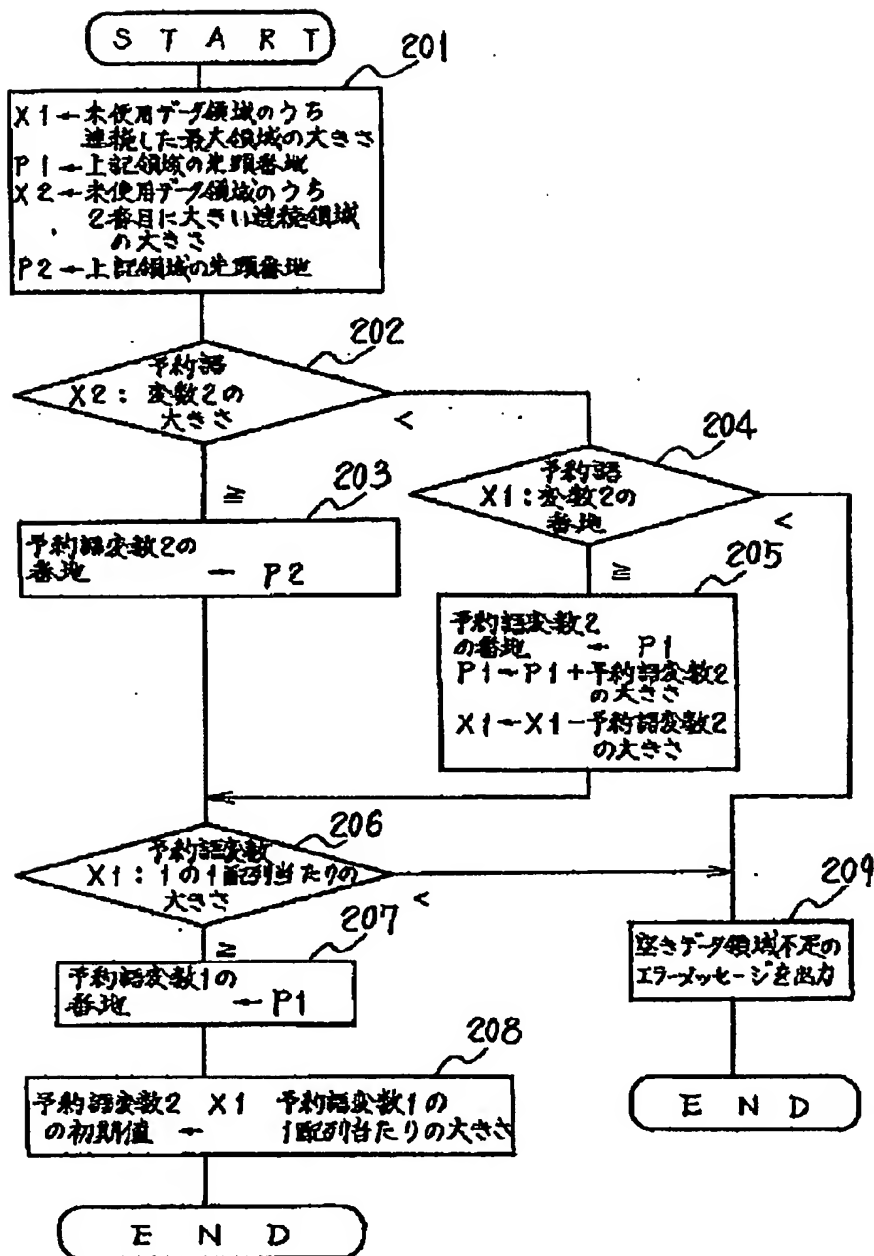
11, 12---オブジェクト

13---シンボル・テーブル

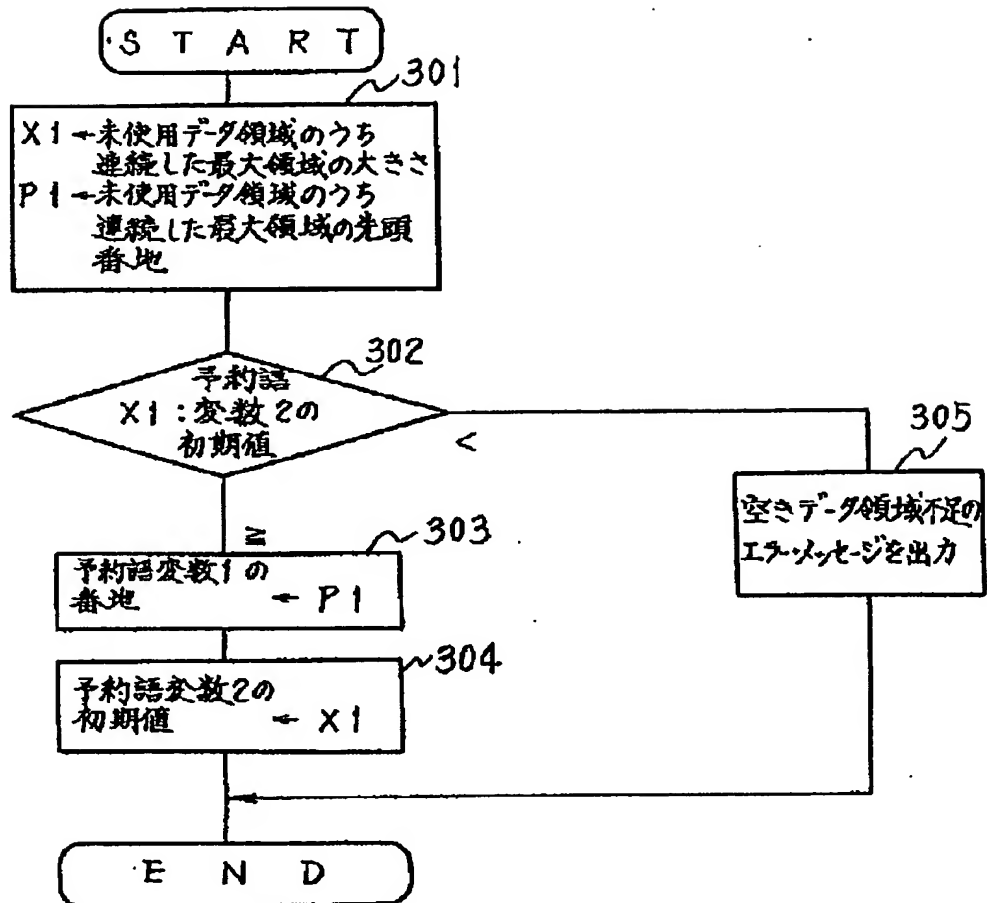
14---実行可能プログラム

15---マップ・リスト

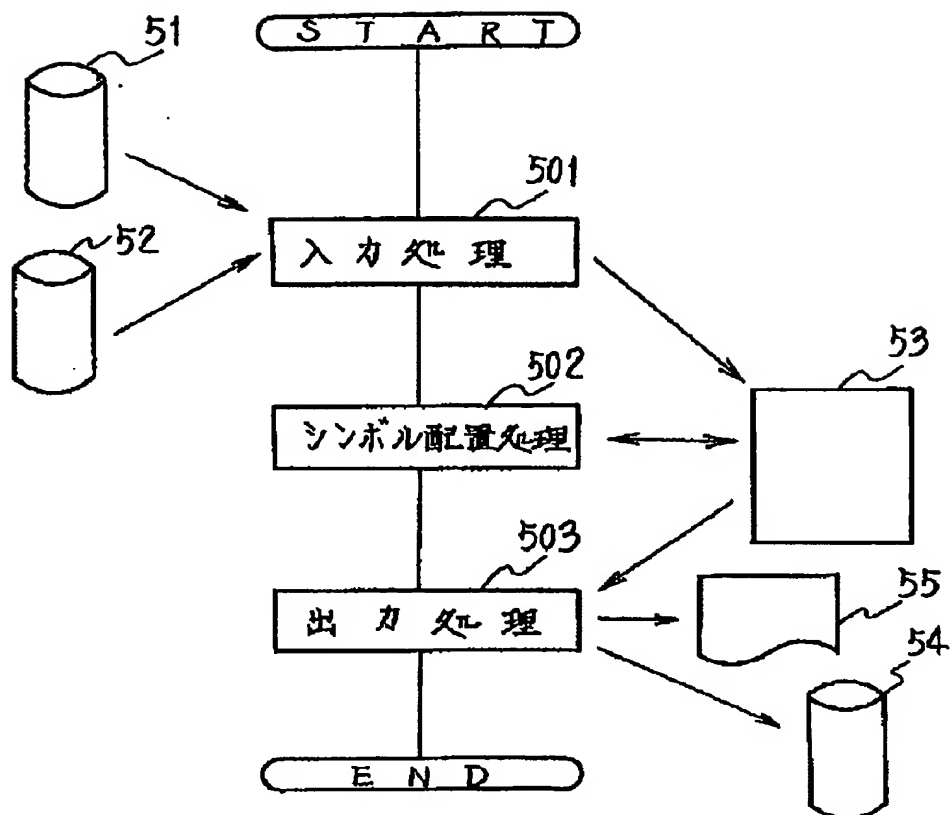
【図2】



【図3】

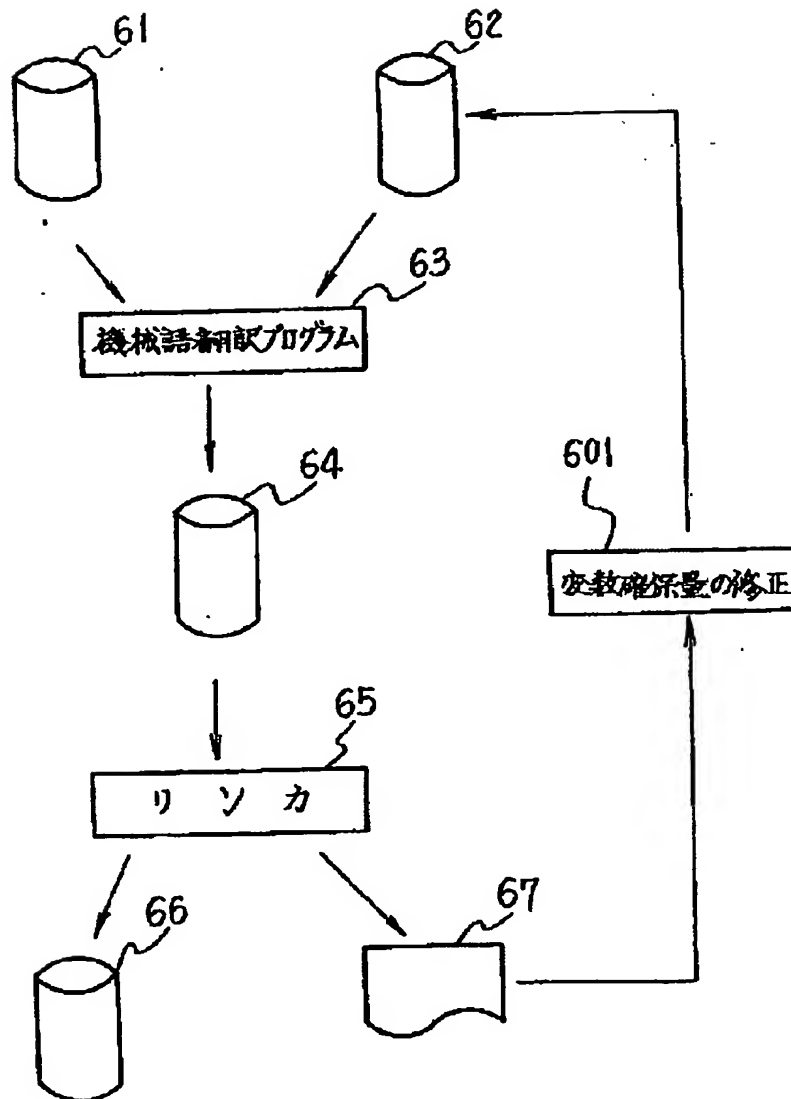


【図5】



51, 52--- オブジェクト 53--- シンボル・テーブル
 54--- 実行可能プログラム 55--- マップ・リスト

【図6】



61, 62---原始プログラム 64---オブジェクト
 66---実行可能プログラム 67---マップ・リスト